Algorithm Spring made by Jurijus Pacalovas compression by calculus and reverse: Size of file should bigger than 1970 bytes before when want you to compress. Size is blocked 1970 bytes. bnk=bnk*255 ghjd=ghj*bnk cvz=cvz+ghdj

bnk=bnk*256 ghjd=ghj*bnk cvz=cvz+ghdj

make smaller than size when lenfa<=14305 bits and save size 14307 bits when bigger save size 14320. When size 14320 bits take **0000...111...'*0X or* *111...0X move to the end* 0000...111...**0 or **111...**0 and change last one to 0. make them together from right to left. Cout this size and save it in bytes than the count of this long and again of this long that should be 1 byte. When sizing 14305 bits save as 1111...0 and size will become 14307 bits.

left 0x and 1111....0x

Save how many times was compressed by one byte.

check: if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and lenfg>0 and cvb==0: lenfg mean when the size of data does not exist information on the block and save information about the first information not exist and change 255 to information not exist. if lenf1<=sssssw or sssssw<=2000 or qqqwz==255 or cvb==1: check the size of the file and check 255 and cvb mean check when the size not if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and lenfg>0 and cvb==0: and save this file.

import binascii a=0 cvb=0 zsaqq="" qqqwz=0 assx=0 ass=0 asss=0 b=0 aaqw="" aaqws="" l="" j=0 b=0 aq=0 qfl=0 t=0 h=0 byteb="" notexist="" lenf=0 numberschangenotexistq = [] numberschangenotexistqz = [] qwa=0 m = [] p=0 namea="" d=1 a=0 asd="" b=0 szx="" asf2="0b" while b<1790:

```
m+=[-1]
b=b+1
```

k = [] wer="" numberschangenotexist = [] numbers = [] name = input("What is name of file? ") namea="file.Spring" namem=name+"/" s="" with open(namea, "w") as f4:

```
    f4.write(s)
```

with open(namea, "a") as f3:

```
    f3.write(namem)
```

with open(name, "rb") as binary_file:

```
# Read the whole file at once
data = binary_file.read()
s=str(data)
lenf1=len(data)
while assx<10:
    if qqqwz>1:
        lenf1=sssssw
    a=0
    ass=0
    asss=0
    b=0
    aaqw=""
```

```
aaqws=""
l=""
j=0
b=0
aq=0
qfl=0
t=0
h=0
byteb=""
notexist=""
lenf=0
numberschangenotexistq = []
numberschangenotexistqz = []
qwa=0
m = []
p=0

d=1
a=0
asd=""
b=0
szx=""
asf2="0b"
while b<1790:
    m+=[-1]
    b=b+1
k = []
wer=""
numberschangenotexist = []
numbers = []

s=""

if lenf1<2000:
    print("This file is too small");
    raise SystemExit
if lenf1>2**30:
    print("This file is too big");
    raise SystemExit
with open(namea, "ab") as f2:
    for byte in data:
        av=bin(byte)
        a=a+1
        if a<=1790:
            byte=int(byte)
            m[byte] = byte
            numbers.append(byte)
            h=h+1

        if a == 1790:
            p=0
            while p<256:
                if p!=m[p]:
                    k.append(p)

                p=p+1

            #lenf count
            lenfg=len(k)


            if lenfg>0:

                notexist=k[0]
```

```
b=-1
bb=0
kl=1789
bnk=0
cb=0


bb=-1
er=-1
ghj=0
ghjd=1
bnk=1
p=-1
cvz=0
qwa=qwa+1
while p<1789:
    p=p+1
    if lenfg>0:
       if 255!=numbers[p]:
            byteb=numbers[p]
            numberschangenotexist.append(byteb)
       if 255==numbers[p]:
           numberschangenotexist.append(notexist)
    if lenfg==0:
        byteb=numbers[p]
        numberschangenotexist.append(byteb)

    #count 1789





    ghj=numberschangenotexist[p]
    qfl=qfl+1
    ghjd=ghj
    bnk=1
    bnks=1
    bb=-1
    bnkd=1


    kl=kl-1
    if qwa<=1:
        while bb<kl:
            if qwa<=1:
                bb=bb+1


            if qwa<=1:
                bnk=bnk*255


            if qwa<=1:
                bnks=bnks*256



    if qwa<=1:
```

```
                numberschangenotexistq.append(bnk)
                numberschangenotexistqz.append(bnks)
            if lenfg>0:
                bnk=numberschangenotexistq[p]
                ghjd=0
                ghjd=ghj*bnk

            if lenfg==0:
                bnks=numberschangenotexistqz[p]
                ghjd=0
                ghjd=ghj*bnks
            cvz=cvz+ghjd
        szx=bin(cvz)[2:]
        lenfa=len(szx)
```

```
        if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and
lenfg>0 and cvb==0:
            cvb=0
        else:
            jl=data
            if cvb==0:
                f2.write(jl)
            cvb=1

        if lenfa>14305:
            wqwe=""
            p=0
            aaqq=""
            d=1
            a=0

            while p<2:

                aaqq=szx[a:d]

                if aaqq=="1":
                    a=a+1
                    d=d+1
                    aaqq=str(aaqq)
                    aaqw=aaqw+aaqq
                if aaqq=="0":
                    p=2

            aaqwss=len(aaqw)
            aasqq=""
            ass=0
            asss=0

            ass=aaqwss
            asss=aaqwss-1
```

```
    aaad="0"
    aasqq=szx[0:asss]
    aasqq=str(aasqq)
    szx=szx[d:]



    aaqw=""
    zzaax=""
    xc=14320-lenfa
    z=0
    if xc!=14320:
        while z<xc:
            zzaax="0"+zzaax
            z=z+1
    wer=wer+szx
    aaqws=aaqws+zzaax+aasqq+"0"
    szx=""
    zzaax=""
    lenf=len(szx)
    wqwe=""
    wqwe=szx[0:1]
    if wqwe=="1":
        raise SystemExit

if lenfa<=14305:
    szx="0"+szx
    xc=14306-lenfa
    z=0
    if xc!=14306:
        while z<xc:
            szx="1"+szx
            z=z+1
    wer=wer+szx
    lenf=len(szx)
    szx=""
    if lenfg>0:
        notexist=k[0]
        szx=bin(notexist)[2:]
        lenf=len(szx)

        xc=8-lenf
        z=0
        while z<xc:
            szx="0"+szx
            z=z+1
        wer=wer+szx
        szx=""



a=0
numberschangenotexist = []
del k[:]

del numbers[:]
m = []
b=0
while b<1790:
    m+=[-1]
    b=b+1
b=0
```

```
        b=0


    s=h%1790
if s!=0:

    s=s-1
    p=-1
    if s!=1789:
        b=-1
        bb=0
        kl=s
        bnk=0
        cb=0
        er=0

        bb=-1
        cvz=0
        ghj=0
        ghjd=1
        bnk=1
        while p<s:
            p=p+1
            byteb=numbers[p]
            numberschangenotexist.append(byteb)

                    #count 1789










            ghj=numberschangenotexist[b]
            ghjd=ghj
            bnk=1
            bb=-1
            kl=kl-1
            while bb<kl:
                bb=bb+1
                bnk=bnk*256
            ghjd=0
            ghjd=ghj*bnk
            cvz=cvz+ghjd
        szx=bin(cvz)[2:]
        lenf=len(szx)


        ert=0
        s=s+1
        ert=s*8



        szx=""
```

```
a=0
szx=""



dd=len(aaqws)


szxzzz=""
szxzzz=bin(dd)[2:]
dd=len(szxzzz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxzzz="0"+szxzzz
        z=z+1

dd=len(szxzzz)



szxz=bin(dd)[2:]
dd=len(szxz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxz="0"+szxz
        z=z+1

dd=len(szxz)
szxzz=""

szxzz=bin(dd)[2:]
dd=len(szxzz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxzz="0"+szxzz
        z=z+1

wer="0b1"+wer+aaqws+"1"
szx=""

lenf=len(wer)
xc=8-lenf%8
z=0
if xc!=8:
    while z<xc:
        szx="0"+szx
        z=z+1

wer=wer+szx+szxzzz+szxz+szxzz
szx=""

n = int(wer, 2)
jl=binascii.unhexlify('%x' % n)
data=jl
sssssw=len(jl)
qqqwz=qqqwz+1
```

```
if lenf1<=ssssssw or ssssssw<=2000 or qqqwz==255 or cvb==1:
    if cvb==1:
        qqqwz=qqqwz-1
    szx=bin(qqqwz)[2:]
    lenf=len(szx)
    xc=8-lenf%8
    z=0
    if xc!=8:
        while z<xc:
            szx="0"+szx
            z=z+1
    zsaqq="0b"+zsaqq+szx

    szx=""

    n = int(zsaqq, 2)
    jlz=binascii.unhexlify('0%x' % n)
    assx=10
    if assx==10 and cvb==0:
        f2.write(jl)
        f2.write(jlz)
    if assx==10 and cvb==1:
        f2.write(jlz)
```